논문 21-46-12-04 The Journal of Korean Institute of Communications and Information Sciences '21-12 Vol.46 No.12
https://doi.org/10.7840/kics.2021.46.12.2144

한국어 랩 가사 생성을 위한 운율 단어 임베딩과 어텐션

박 찬 솔•, 정 윤 경°, 이 종 현*

Rhyme Word Embedding and Attentionfor Korean Rap Lyrics Generation

Chansol Park•, Yun-Gyung Cheong°, Jong-Hyun Lee*

요 약

가사 생성 연구는 음악 창작 활동에 많은 도움이 되고 있다. 하지만 힙합이라는 특정한 장르의 랩 가사는 단순히 가사의 의미뿐만 아니라 운율의 구성도 요구된다. 따라서 운율이 구성된 한국어 랩 가사를 생성하기 위해서는 한국어 음절을 고려한 가사 생성 연구가 필요하다. 한국어 음절은 영어와 다르게 2~3개의 자음과 모음으로 이루어져 있다. 이러한 단어의 구조적 차이 때문에 한국어 가사 생성은 그에 맞는 적합한 모델이 필요하다. 특히, 랩 가사는 문맥 정보뿐만 아니라 운율 정보를 포함하고 있어 이에 대한 고려가 필요하다. 본 논문에서는 앞의 두 정보를 결합한 단어 임베딩 모델을 사용하여 한국어 운율에 적합한 랩 가사 생성 모델을 제안한다. 한국어 단어는 각 글자마다 하나의 음절로 되어있으며 각 음절은 문자 단위인 초성, 중성, 그리고 종성으로 구성되어 있다. 이러한 음절 특성으로부터 운율 정보를 얻기 위해 단어를 문자 단위로 재배열한 후 운율 정보를 갖는 하위 단어들로 단어를 재구성하여 사전 훈련하는 운율 단어 임베딩 모델을 제안하였다. seq2seq 형태의 가사 생성 모델을 학습하기 위해 인코더-디코더 모델을 설계하였고, 입력 문장과 출력 문장 사이의 운율 관계를 갖는 특정 단어 쌍에 주목하기 위해 어텐션 메커니즘을 사용하였다. 마지막으로, 제시된 검증 방법을 이용하여 운율을 고려한 한국어 가사 생성 모델과 기존의 문맥 정보만을 사용하는 모델을 비교하여 우수성을 확인하였다.

Key Words : Rhymes, Rap Lyrics, Korean Syllables, Word Embedding, Natural Language Processing

ABSTRACT

Rap lyrics require not only the meaning of the lyrics, but also the composition of the rhyme. In order to generate Korean rap lyrics composed of rhymes, it is necessary to study lyrics generation considering Korean syllables. Korean syllable consists of two or three combination of consonant and vowel characters. In this respect, Korean lyric generation model has a specific structure different from English-based lyric generation model. Furthermore, for Korean rap lyric generation model, rhyme information should be included in its structure because rap lyrics includes rhyme information as well as context information. In this paper, a rap lyrics generation model using the embedding model combining these two informations was proposed. To implement the embedding model, each syllable of a Korean word was split into initial consonant, medial (vowel), and final consonant. Then, they were rearranged into characters and again re-organized to subwords by grouping characters including rhyme information. To learn a seq2seq type of lyrics generation model, an

※ 본 연구는 과학기술정보통신부의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2020R1C1C1009720)
• First Author : Sungkyunkwan University Department of Electrical and Computer Engineering, csglanc@skku.edu, 정회원
° Corresponding Author : Sungkyunkwan University Department of Software, aimecca@skku.edu, 정회원
* Sungkyunkwan University Research Center for Convergence, jhlee@creativemind.ai
논문번호 : 202109-259-A-RU, Received September 29, 2021; Revised October 7, 2021; Accepted October 7, 2021

encoder-decoder model was designed and attention mechanism was used to capture the specific word pair with a rhyme relationship between the input and output sentences. Finally, we performed performance evaluation of the proposed rap lyrics generation model and confirmed the better performance than that of the conventional model using only context information.

Ⅰ. 서 론

인공 지능의 응용 분야가 확대됨에 따라 창작을 위한 텍스트 생성에 인공 지능을 가미한 많은 연구가 진행되고 있으며 그중 하나가 가사 생성이다. 가사 생성을 위한 기존의 연구는 대부분 앞뒤 가사와의 의미전달에 초점을 맞추었다[1]. 그러나 힙합 장르의 랩 가사는 가사의 의미뿐만 아니라 운율의 구성도 요구된다.

운율 정보를 포착하고 가사에 적용하기 위한 선행 연구로는 영어 가사의 운율 관계 유사성을 고려한 랩 가사 생성 사례들이 있다[2,3]. 하지만 영어는 한국어와 다른 언어 구조를 갖고 있어 한국어 랩 가사 생성을 위해서는 이에 대한 다른 접근 방식이 필요하며, 이러한 한국어 단어의 구조적 특징을 반영한 연구 사례는 한국어 운율 분석에 적합한 방법을 보여 준다[4].

한국어 음절은 영어와 다르게 2~3개의 자음과 모음으로 이루어져 있어 한국어 랩 가사 생성을 위해서는 한국어 단어의 자음과 모음 체계에 대한 이해와 운율 정보와의 연관성을 고려하는 것이 필요하다.

가사(또는 텍스트) 생성에 관한 연구들은 문장의 문맥을 고려하기 위해 문장의 앞뒤 이웃 단어를 벡터로 밀집 표현(dense representation)하는 방식을 택하고 있다[5]. 벡터를 입력으로 하는 RNN(Recurrent Neural Network) 기반의 언어 모델은 n-gram 단어 임베딩 모델보다 우수한 성능을 보여주나 단순한 형태의 RNN은 타임 스텝이 길어질수록 이전 스텝의 은닉된 상태를 제대로 전달하지 못하는 기울기 손실(vanishing gradient)이 발생한다[6].

이러한 문제를 극복하고자 가사 생성을 위한 LSTM(Long Short-Term Memory) 모델이 제안되었다[1,2]. 인코더와 디코더의 신경 네트워크 형태를 갖는 seq2seq (sequence-to-sequence) 모델의 경우 LSTM 셀을 이용하여 인코더와 디코더를 구성할 수 있으며, 인코더는 입력 데이터 시퀀스(문장)을 임베딩된 벡터를 통해 인코딩하고 디코더는 타겟 시퀀스(문장)을 디코딩 함으로써 입력 시퀀스와 출력 시퀀스 사이의 단어 정렬에 집중 할 수 있도록 하여 신경 기계 번역 뿐만 아니라 문장 생성에도 탁월한 성능을 보여 준다[7,8].

다만, seq2seq 모델은 고정된 문장 임베딩을 사용하기 때문에 문장이 길어질수록 많은 양의 정보를 다 담지 못하고 손실되는 단점을 갖고 있어 이에 대한 보완으로 어텐션 메커니즘 방식이 사용되었다[9,10]. 이는 인코더에서 주어진 시퀀스(문장)의 길이와 관계없이 은닉 상태의 모든 정보를 디코더에 넘겨주어 매 타임 스텝마다 인코더의 모든 정보를 다시 한 번 참고할 수 있도록 해주어 연관성이 높은 정보에 더 주목할 수 있고 이를 학습에 반영할 수 있어 랩 가사의 특징을 포착하고 학습하는데 적합하다[11,12].

현재 이러한 어텐션 메커니즘은 다양한 형태로 가사 생성 모델에 널리 사용된다. 어텐션을 기반으로 하는 중국어 노래 생성 모델이 있으며[13], 인코더-디코더 형태의 일본어 가사 생성 모델도 있다[14]. 또한 음절과 부합되는 멜로디 정보와 문맥 정보를 모두 고려하기 위해 다채널(multi-channel) seq2seq 모델이 제안되었고[15], 특정 작사가의 패턴이나 스타일을 학습하기 위해 문장과 문서 수준의 문맥 정보 포착을 위한 계층적 어텐션 모델도 보고된 바 있다[16]. 지난 연구에서 우리는 한국어로 된 랩 가사 생성 모델에 기초가 되는 운율 정보를 포함하는 하위 단어 수준의 벡터 표현 방식을 제안하였다[17].

본 논문에서는 이러한 선행 연구들을 토대로 한국어 단어의 음절 정보를 고려한 단어 임베딩 모델을 제안하고, 해당 모델을 LSTM으로 구성된 인코더-디코더에 연결하여 한국어 라임 정보를 고려한 가사 생성 모델을 구성하며, 해당 모델 학습 시 주어진 입력 문장과 다음에 올 타겟 문장 사이의 특정 단어 짝의 운율 관계에 집중할 수 있도록 어텐션 메커니즘을 적용하고, 성능 평가 모델을 통해 검증한다.

Ⅱ. 제안 기법: 운율 단어 임베딩과 어텐션

가사 생성을 위한 자연어 처리를 위해서는 벡터방식의 단어 표현이 필요하며, 운율 정보를 포함하기 위해 본 논문에서는 두 가지 표현 방식을 제안한다. 하나는 가사의 의미를 전달하기 위한 '문맥 표현(Representation of Context)'이고 다른 하나는 운율을 갖는 랩 가사를 위한 '운율 표현(Representation of Rhyme)'이다.

2.1 문맥 표현

가사에 있어서 한 단어의 의미는 그 단어의 전후 문맥에 따라 달라질 수 있다. 특히 모호한 표현이 많은 가사일수록 문맥의 흐름으로 단어의 의미를 전달하는 것이 중요하다. 이는 양방향 언어 모델을 기반으로 한 ELMo 단어 임베딩을 사용하여 구현할 수 있으며, 단어의 앞부분만 아니라 뒷부분의 문맥에 따라 다른 의미로 전달될 수 있다[6].

2.2 운율 표현

운율 표현을 위한 첫 번째 과정은 영어와 다른 발음 구조를 가진 한국어 음절의 운율 정보를 벡터화 하는데 있다. 그림 1은 한국어 단어의 예를 보여준 것으로, 한 음절은 한 글자로 이루어져 있으며, 하나의 글자는 초성, 중성(모음)과 종성 세 개의 자음과 모음으로 구성된다.

그림 2는 '단어'와 '연어' 두 단어의 유사성을 보여준 것으로 두 단어의 초성, 중성, 종성 중 절반 이상이 같음을 알 수 있다. 이런 특성을 추출하기 위해 단어를 글자 단위로 나누고 각 글자를 다시 문자 단위로 나눈 다음 각 글자의 동일한 위치(즉, 초성과 중성 그리고 종성)을 그룹으로 묶어 하위 단어를 만들었다.

특히, 앞 음절의 종성과 다음 음절의 초성이 단어의 운율을 결정하는데 연관되어 있어, 이들도 하위 단어 그룹에 포함시켰으며, 마지막으로 하위 단어에 대한 임베딩 개념을 사용하기 위해 Fasttext 단어 임베딩 모델을 적용하였다[18].

표 1은 '자연어'라는 단어로부터 하위 단어들을 얻는 과정을 보여준다. 여기서 '-'는 자음이 없는 경우를 의미한다. 먼저 '자'의 글자를 'ㅈ', 'ㅏ', '-'의 세 요소로 나누고, 같은 방법으로 '연'은 'ㅇ', 'ㅕ', 'ㄴ'으로, '어'는 'ㅇ', 'ㅓ', '-'로 나누었다. 이를 바탕으로 각 글자의 초성, 중성 또는 종성끼리 모아 하나의 하위 단어를 구성하는 집합을 얻었다. 예를 들어, 초성에 해당되는 요소를 모으면 'ㅈ', 'ㅇ', 'ㅇ'의 집합이 얻어진다. 초성, 중성, 종성에 대한 운율 정보는 하위 단어를 구성하는 집합 안에 각각 '#', '$', '%'를 추가하여 구분하였다. 이외에도 문자와 문자사이의 종성 및 초성 관계는 음절 정보에 영향을 미치므로 이를 반영하기 위해 식별기호 '&'와 함께 <&, -, ㅇ>, <&, ㄴ, ㅇ>을 추가하였다.

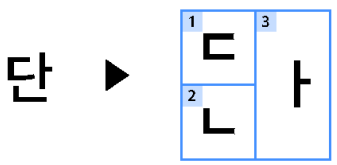

그림 1. '단'의 초성, 중성, 종성
Fig. 1. 1) initial consonant, 2) medial(vowel), and 3) final consonant of the word '단'

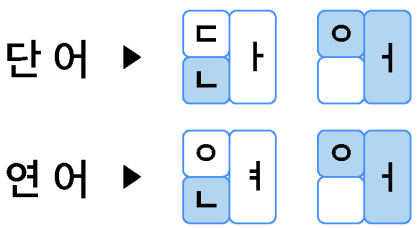

그림 2. 유사한 운율을 갖는특 두 단어
Fig. 2. Two words with similar rhymes

표 1. '자연어'에 대한 하위 단어 구성 예
Table 1. Example of subwords for the word '자연어'

Levels	Elements
Word	자연어
Letters	자, 연, 어
Characters	ㅈ, ㅏ, -, ㅇ, ㅕ, ㄴ, ㅇ, ㅓ, -
Subwords	〈#, ㅈ, ㅇ, ㅇ〉, 〈#, ㅈ, ㅇ〉, 〈#, ㅇ, ㅇ〉, 〈#, ㅈ, ㅇ〉, 〈$, ㅏ, ㅕ, ㅓ〉, 〈$, ㅏ, ㅕ〉, 〈$, ㅕ, ㅓ〉, 〈$, ㅏ, ㅓ〉, 〈%, -, ㄴ, -〉, 〈%, -, ㄴ〉, 〈%, ㄴ, -〉, 〈&, -, ㅇ〉, 〈&, ㄴ, ㅇ〉

2.3 두 표현의 연결

일반적으로 랩 가사에는 문맥과 운율이라는 두 가지 정보를 포함하고 있어 랩 가사를 생성하기 위한 자연어 처리를 위해 두 정보를 모두 포함하는 단일 단어

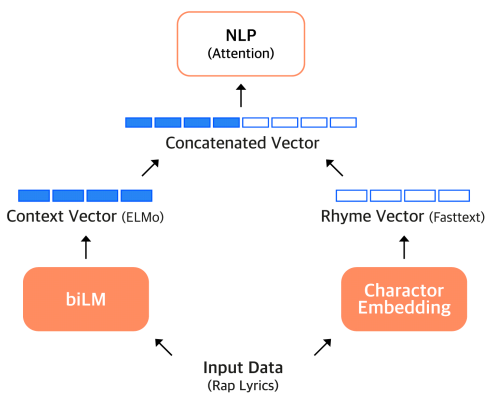

그림 3. 문맥과 운율정보로 구성된 하나의 임베딩 벡터
Fig. 3. A word embedding vector including context and rhyme

임베딩 모델이 필요하다. 이는 다음과 같이 두개의 단어 임베딩 벡터를 연결하여 해결하였다. 그림 3은 문맥과 운율의 임베딩 벡터를 상호 연결시켜 생성한 단어 표현 과정을 보여준다.

2.4 가사 생성을 위한 Seq2Seq 모델

랩 가사에서 운율 관계는 각 줄(문장)의 비슷한 단어 위치에 잘 표현되어 있다. 예를 들면, 각 줄(문장)의 마지막 단어에서 라임 관계가 자주 형성되는 것을 볼 수 있는데 기존의 LSTM만을 적용한 랩 가사 생성 모델은 입력된 줄(문장) 전체를 고려하지 않고 바로 이전의 은닉 상태만을 고려하기 때문에 주어진 문장에 잘 대응하는 문장을 생성하기 어렵다[2]. 이를 보완하기 위해 seq2seq 형태의 모델 구조를 도입하여 입력 문장의 압축된 정보를 문맥 벡터로 디코더에 전달하여 입력 문장 전체를 고려한 문장을 생성할 수 있도록 하였다[7]. 다만, 가사 생성 학습을 위한 입력 데이터는 가사에 따라 그 길이가 다양한데 반해 seq2seq 모델은 고정된 차원의 벡터를 인코더에서 디코더로 전달하므로 문장이 긴 경우 필요한 모든 정보를 담지 못하게 되어 문장 생성 시 다음에 나올 단어를 예측하는 과정에서 입력 줄(문장) 전체에 해당하는 모든 단어 정보를 참고하기 어렵다. 이러한 문제를 해결하기 위해 모델 학습 시 서로 관련이 깊은 특정 단어들 간의 운율 관계 정보에 더 주목할 수 있도록 어텐션 메커니즘을 사용하였다[9].

2.5 운율 자리의 포착

그림 4는 본 논문에서 제안하고자 하는 모델 구조를 보인 것으로, 입력된 각 단어에 대한 모든 은닉 상태를 하나로 결합시켜 디코더의 어텐션 계층으로 보내고, 입력된 줄(문장)의 각 단어에 해당하는 모든 LSTM의 은닉 상태를 문맥 벡터로 만들어 가사의 줄(문장)들 간에 유사한 운율 관계를 갖도록 하였다.

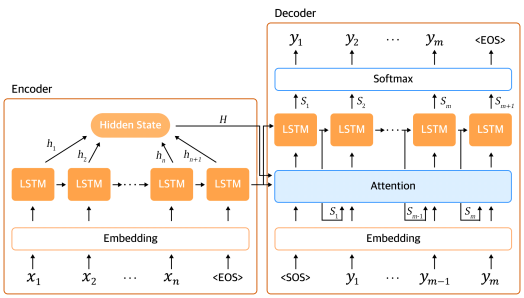

그림 4. 제안하고자 하는 자연어처리 모델의 구조
Fig. 4. The structure of proposed NLP model

즉, 출력에 대한 현재시점을 t라고 할 때, 어텐션 메커니즘은 입력 줄(문장)에 대한 인코더의 은닉 상태와 디코더의 $t-1$ 시점의 은닉 상태를 모두 활용한다. 이렇게 함으로써 출력 단어를 예측하는 t 시점에서 입력된 라인(문장)의 어느 부분에 더 주목해야 하는지에 대한 가중치를 주는 문맥 벡터를 얻을 수 있다. 현시점 t 에서의 디코더의 은닉 상태를 S_t라 하고 출력 단어를 Y_t라 하면 랩 가사 생성을 위한 $\hat{Y}$의 확률 모델은 다음과 같다.

$$\hat{Y} = p(Y_t | Y_0, \ldots, Y_{t-1}, X) \tag{1}$$

$$= Softmax(W_y S_t + b_y) \tag{2}$$

여기서 X는 입력 줄(문장)의 단어들로 이루어진 집합을 그리고 W_y와 b_y는 Y에 대한 가중치 행렬과 편향(bias)을 의미한다. LSTM 계층에서 $Softmax$ 출력 계층으로 보내기 위한 S_t는 다음과 같다.

$$S_t = \tanh(W_{ys} Y_{t-1} + W_{ss} S_{t-1} + W_{cs} C_t + b_s) \tag{3}$$

여기서 C_t는 다음에 언급될 현시점에서 디코더의 어텐션 값에 대한 문맥 벡터를 의미하며, W_{ys}, W_{ss}와 W_{cs}는 각각 W, S 그리고 C에 대한 가중치 행렬이고, b_s는 S에 대한 편향이다. 어텐션 계층에서 이전 출력 단어와 이전 은닉 상태 그리고 문맥 벡터를 더한 후, 활성화 함수인 하이퍼볼릭 탄젠트 함수를 거치면 현 시점에서의 최종 S_t가 얻어진다. 또한, 어텐션 가중치가 반영된 C_t는 다음과 같다.

$$C_t = \sum_{j=1}^{n+1} \alpha_j^t h_j \tag{4}$$

여기서 h_j는 j번째 단어에 대한 인코더의 은닉 상태를 n은 집합 X의 단어 수를 의미하며, α_j^t는 어텐션 계층에서 문맥이나 운율과 같은 특정 정보에 대한 어텐션 가중치이다. t시점에서의 C_t는 모든 n에 대해 은닉 상태 h_j에 상응하는 어텐션 가중치 α_j^t $(j = 1 \ldots n+1)$의 가중합으로 표시되며, 어텐션 가중치 α_j^t는 S_{t-1}과 h_j의 어텐션 스코어를 통해 결정된다. 어텐션 스코어 함수는 concat 스코어 함수가 사용되었고 해당 식은 다음과 같다[9].

$$Score(S_{t-1}, h_j) = v_s^T \tanh(W_s S_{t-1} + U_s h_j) \quad (5)$$

여기서 W_s, U_s그리고 v_s는 스코어에 대한 학습 가능한 가중치 행렬을 의미한다. 주어진 스코어 함수로부터 t시점에서 이전 단계의 디코더 은닉 상태 S_{t-1}과 현재 인코더의 모든 은닉 상태 H와의 유사성을 구할 수 있다. 이와 같은 방식으로 주어진 입력 데이터의 어떤 정보(단어)에 더 주목해야 하는지 단계별로 파악할 수 있다.

Ⅲ. 실험 및 평가

3.1 데이터셋

실험 방법을 위해 기존의 학습 데이터셋을 이용할 수 있으나 운율이 보다 많이 포함된 가사 데이터를 확보하기 위해 무작위 랩 가사 데이터 수집이 아닌 국내 힙합 영역에서 인지도가 높은 100인의 래퍼들에 대한 가사를 수집하였다. 국내 음원 유통 사이트를 통해 특정 래퍼들의 랩 가사 데이터를 크롤링하여 총 4,539개의 가사 데이터를 수집하였다.

수집 데이터로부터 문맥 벡터를 가지는 단어 임베딩 모델과 운율 벡터를 가지는 단어 임베딩 모델을 학습하였으며, 운율 벡터를 가지고 있는 단어 임베딩 모델의 성능이 입증되었다는 가정 하에 가사 데이터의 첫 번째 줄(문장)과 두 번째 줄(문장) 사이의 운율 점수를 계산하였다. 운율 임베딩 모델의 성능 평가 방법과 운율 임베딩 모델을 이용한 두 줄(문장) 사이의 운율 점수 계산은 평가 섹션에서 다룬다. 이와 같은 방법으로 우리는 전체 가사 줄(문장) 데이터로부터 운율 관계를 가지는 줄(문장)의 짝을 엄선하였고 총 24,959개의 운율 줄(문장)의 짝을 얻었다. 22,459의 줄(문장) 짝을 트레이닝 데이터로 사용하였으며 2500개의 줄(문장) 짝을 테스트 데이터로 사용하였다.

3.2 실험 설정

신경망 모델의 실험 설정은 256차원의 임베딩 벡터, 256차원의 은닉 상태, 4계층으로 된 LSTM 네트워크로 진행하였다. 학습 트레이닝을 위한 Dropout의 비율은 0.3, batch 크기는 256, epoch 수는 100이다. 어텐션 메커니즘으로 Bahdanau Attention을 사용하였으며 손실 함수는 binary cross-entropy를 최적화를 위해서는 Adam 옵티마이저를 사용했다.

3.3 가사 생성

랩 가사 생성 모델의 성능을 테스트하기 위해 주어진 가사 줄(문장)으로 새로운 랩 가사 줄(문장)을 생성하였다. 학습한 seq2seq 형태의 자연어처리 모델을 통해, 테스트 데이터셋인 운율 줄(문장) 짝의 첫 번째 줄(문장)로부터 그 다음에 올 가사 줄(문장)을 생성하도록 하였으며, 생성된 줄(문장)을 테스트 데이터셋인 운율 줄(문장) 짝의 두 번째 줄(문장)과 비교 평가하였다.

3.4 운율 유사성 평가

운율 유사성 평가 관련해서 영어 운율에 대한 점수 행렬을 만들어 두 단어 간의 운율 점수를 구하는 방식이 보고된 바 있다[19]. 이러한 운율 점수 방법은 생성된 영어 랩 가사에 대한 운율 평가는 가능하나 운율 점수 매트릭스가 영어 단어에 맞춰져 있어 한국어 단어로 된 랩 가사에는 적합하지 않다[20].

본 논문에서는 운율 평가 모델로서 앞서 제시한 운율 벡터 임베딩 모델을 사용하며 제안한 모델이 두 단어사이의 운율 유사성을 잘 잡아내는지에 대한 성능 검증을 위해 두 가지 유형의 라임 성능 평가 문제를 직접 만들었다.

첫 번째 문제는 세 개의 단어를 보여주고 운율 관계가 없는 하나의 단어를 찾는 문제이며, 두 번째 문제는 세 개의 단어 선택지와 하나의 평가 단어를 보여주고 평가 단어와 운율 유사도가 높은 단어순으로 순서를 맞추는 문제이다. 30개의 첫 번째 유형 문제와 20개의 두 번째 유형 문제를 합해 총 50개의 라임 성능 평가 문제를 만들었다. 그리고 사람이 제시한 문제의 정답과 운율 모델이 제시한 문제의 정답을 비교하여 운율 단어 임베딩 모델의 성능을 평가하였다.

3.5 가사 생성 평가

랩 가사 생성의 성능을 평가하기 위해 가사 앞줄(문장)의 단어들과 뒷줄(문장)의 단어들 사이의 운율 유사성을 계산하였다. 운율 유사성은 운율 단어 임베딩 모델이 두 단어 사이의 운율 관계를 잘 파악한다는 가정 하에 해당 모델의 코사인 유사도 함수를 이용해 결정할 수 있으며, 두 단어 사이의 운율 유사성은 다음과 같은 코사인 거리(cosine distance) 함수를 사용하였다.

$$S(w_i, w_j) = \cos(\theta) = \frac{v_i \cdot v_j}{|v_i||v_j|} \quad (6)$$

여기서 운율 유사성 점수는 0과 1 사이의 값을 갖는다(1에 가까울수록 유사성이 더 높다). 가사의 구절에 해당하는 모든 줄(문장) 끝에 있는 단어 사이의 운

율 점수를 합산하여 최종 점수가 얻어지며 종합 운율 점수(TRS: Total Rhyme Score)는 다음과 같다.

$$TRS = \frac{1}{n}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} S(w_i, w_j) \quad (7)$$

여기서 w_i는 i번째 줄(문장)의 마지막 단어를 의미하고, n은 가사 줄(문장)의 수를 의미하며, $S(w_i, w_j)$는 w_i와 w_j 사이의 운율 유사성에 대한 점수 함수를 의미한다. 문맥 정보만을 가지는 단어 임베딩을 사용한 자연어 처리 모델과 본 논문에서 제안한 모델을 비교하여 운율 정보의 유무에 따른 한국어 랩 가사 생성의 성능 차이를 평가 하였다. 각 모델 별로 생성한 한국어 랩 가사들의 종합 운율 점수들을 구하고, 이를 토대로 가사 내 단어들 간의 운율 관계가 얼마나 잘 형성되어 있는지를 비교 분석하였다.

Ⅳ. 결 과

4.1 운율 유사성 결과

앞서 제시한 평가 방법들을 통해 제한한 모델들을 성능 평가하였다. 먼저 제안한 운율 단어 임베딩 모델이 두 단어 사이의 운율 유사성을 판별해 낼 수 있는지를 테스트를 통해 확인하였으며, 주어진 운율 유사도 문제를 테스트하여 사람이 제시한 정답과 운율 단어 임베딩 모델이 제시한 정답과 서로 얼마나 일치하는지를 확인하였다.

총 50 문항의 테스트 문제를 통해 운율 단어 임베딩 모델의 성능을 확인한 결과 제안한 모델은 사람이 제시한 답과 비교하여 50개의 문항 중 42개의 답이 일치하였다. 반면에 문맥의 정보만을 가지고 있는 문맥 워드 임베딩 모델은 50개의 문항 중 12개의 문항만이 일치하였다. 다만 우리가 제시한 운율 단어 임베딩 모델은 'ㅖ' 와 'ㅐ'과 같이 비슷한 발음 역할을 하는 모음들 사이의 운율 유사성은 제대로 잡아내지 못하였다. 이는 한국어 단어로부터 하위 단어들을 구성할 때 비슷한 발음을 가지는 모음들에 대한 추가적인 처리가 필요함을 의미한다. 하위 단어들의 구성 방법에 대한 지속적 연구를 통해 개선될 수 있으리라 본다.

4.2 가사 생성 결과

운율 유사도 평가 과정에서 운율 단어 임베딩 모델이 두 단어 사이의 운율 관계를 구분해낼 수 있음을 확인하였으며 운율 단어 임베딩 모델을 기반으로 제안한 모델이 랩 가사에 적합한 가사를 생성해 내는지를 평가하였다. 문맥 정보만을 가지고 있는 단어 임베딩 모델을 사용하여 학습한 자연어처리 모델과 문맥과 운율 두 가지 정보를 모두 가지고 있는 단어 임베딩 모델을 사용하여 학습한 자연어처리 모델을 가지고 테스트를 진행하였다.

표 2는 제안한 모델을 이용하여 생성한 랩 가사 예제를 보여 준다. 운율 관계가 잘 형성된 가사 줄(문장)을 한 짝으로 하여 세 짝의 테스트 데이터셋을 구성하였으며, 데이터셋에 있는 두 줄(문장) 짝의 첫 번째 줄(문장)을 입력 데이터로 사용하여 두 번째 가사 줄(문장)을 생성하였다. 기존 모델들과 비교하기 위해서 생성된 두 번째 줄(문장)과 입력 데이터로 주었던 첫 번째 줄(문장)과의 종합 운율 점수를 계산하고 테스트 데이터셋의 종합 운율 점수와 비교하였다. 예제에서 보듯이 입력으로 주어진 실제 가사 줄(문장)과 생성된 출력 가사 줄(문장)의 마지막 단어들 사이에 운율 관계가 잘 형성되어 있는 것을 볼 수 있다. 테스트 결과는 표 3에 요약되어 있으며, 제안한 모델이 문맥 정보만을 사용하는 모델보다 종합 운율 점수가 더 높게 나온 것을 확인할 수 있었으며, 테스트 데이터셋의 종합

표 2. 제안된 모델로부터 생성된 랩 가사 예제
Table 2. Examples of rap lyrics generated from the proposed model

Test Dataset	Proposed Model
싹 다 디비 가 가가 빨아 무이소 돈만 챙기고 고마 그마 째이소	싹 다 디비 가 가가 빨아 무이소 마음만 남기고 그냥 닫아 두이소
반전이 없는 게 반전이야 발전이 없는 게 빵점이야	반전이 없는 게 반전이야 노력이 없는 게 환상이야
잡혀서 동네 목욕탕에 갔었어 아빠의 때수건은 정말 아팠어	잡혀서 동네 목욕탕에 갔었어 나의 그녀는 길거리로 나갔어

표 3. 실제 가사와 모델별 생성된 가사의 종합 운율 점수
Table 3. Total rhyme scores of the models and the actual lyrics tested

Models	Total Rhyme Score
Context embedding + attention	0.23
Context & rhyme embedding + attention	0.30
Actual lyrics data from the test dataset	0.35

운율 점수에 근사한 종합 운율 점수를 보여주었다. 다만 세 번째 랩 가사 예제를 보면 입력 줄(문장)과 생성된 출력 줄(문장) 간의 운율 관계가 잘 형성되어 있는 반면 앞뒤 줄(문장) 간의 문맥 형성은 다소 부족한데 이는 자연어처리 모델 설계에 대한 지속적인 연구를 통해 개선될 수 있으리라 본다.

V. 결 론

한국어 랩 가사 생성을 위해 운율 정보를 포함하는 단어 임베딩 방법을 제안하였다. 제안한 방법은 운율 정보 추출을 위해 단어를 자음과 모음으로 나누고 이를 바탕으로 하위 단어의 기본단위를 구성한 후 문맥과 운율 정보를 갖는 벡터로 인코더와 디코더를 구성함으로써 구현 가능하다. 제안한 임베딩 방법을 이용하여 자연어 처리모델을 학습시킴으로써 랩 가사 생성이 가능함을 보였다. 문장 단위의 가사 생성은 선행 줄(문장)에 대응되는 뒷줄(문장)의 생성으로 문장 단위의 가사 생성이 가능한 seq2seq 형태의 모델을 이용하였다.

테스트 결과 제안한 모델이 문맥 정보만을 사용하는 모델보다 종합 운율 점수가 더 높게 나온 것을 확인할 수 있었으며, 테스트 데이터셋의 종합 운율 점수에 근사한 종합 운율 점수를 보여 주었다. 다만 본 문에서 언급한 바와 같이 미세한 발음상의 차이를 갖는 부분에 대해서는 성과가 다소 미흡하여 후속 연구를 통해 한국어 단어의 운율 정보를 보다 정확하게 표현할 수 있는 단어 임베딩 방법을 개선하고, 자연어처리 모델의 설계를 보완하여 줄(문장) 단위의 가사 생성이 아닌 절 단위의 가사 생성이 가능하도록 할 계획이다. 그럼에도, 본 연구를 통해 만들어진 한국 랩 가사 생성기는 한국어 랩 창작 활동을 위한 보조적 도구로써 충분히 활용될 수 있을 것이라 생각한다.

References

[1] S. Son, H. Lee, G. Nam, and S. Kang, "Song-lyrics generation system by deep learning," in *Proc. Annu. Conf. Human and Lang. Technol.,* pp. 570-573, Seoul, Korea, Oct. 2018.

[2] P. Potash, A. Romanov, and A. Rumshisky, "GhostWriter: Using an LSTM for automatic rap lyric generation," in *Proc. 2015 Conf. Empirical Methods in Natural Lang. Process.*, Lisbon, Portugal, Sep. 2015.

[3] E. Malmi, P. Takala, H. Toivonen, T. Raiko, and A. Gionis, "DopeLearning: A computational approach to rap lyrics generation," in *Proc. 22nd ACM SIGKDD Int. Conf.*, San Francisco, CA, USA, Aug. 2016.

[4] S. Park, J. Byun, S. Baek, Y. Cho, and A. Oh, "Subword-level word vector representations for korean," in *Proc. 56th Annu. Meeting of the Assoc. Computational Linguistics*, vol. 1, pp. 2429-2438, Melbourne, Australia, Jul. 2018.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vecto space," *arXiv2013*, arXiv:1301.3781v3.

[6] M. E. Peters, M. Neumann, M. Iyyer, and M. Gardner, "Deep contextualized word representations," in *Proc. 2018 Conf. North Am. Chapter of the Assoc. Computational Linguistics: Human Lang. Technol.*, New Orleans, LA, USA, Jun. 2018.

[7] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *arXiv2014*, arXiv:1409.3215v3.

[8] K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder‐decoder for statistical machine translation," in *Proc. 2014 Conf. EMNLP*, Doha, Qatar, Oct. 2014.

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv2016*, arXiv:1409.0473v7.

[10] M. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv2015*, arXiv:1508.04025v5.

[11] A. Tsaptsinos, "Lyrics-based music genre classification using a hierarchical attention network," *arXiv2017*, arXiv:1707.04678v1.

[12] H. Liang, Q. Li, H. Wang, H. Li, J. Wei, and Z. Yang, "AttAE-RL^2: Attention based autoencoder for rap lyrics representation

learning," in *Proc. WWW '18*, Lyon, France, Apr. 2018.

[13] Q. Wang, T. Luo, D. Wang, and C. Xing, "Chinese song iambics generation with neural attention-based model," *arXiv2016*, arXiv: 1604.06274v2.

[14] R. Takahashi, T. Nose, Y. Chiba, and A. Ito, "Successive japanese lyrics generation based on encoder-decoder model," in *Proc. 9th GCCE*, Kobe, Japan, Oct. 2020.

[15] X. Lu, J. Wang, B. Zhuang, S. Wang, and J. Xiao, "A syllable-structured, contextually-based conditionally generation of chinese lyrics," in *Proc. 16th Pacific Rim Int. Conf. Artificial Intell.*, Cuvu, Yanuca Island, Fiji, Aug. 2019.

[16] X. Wu, Z. Du, Y. Guo, and H. Fujita, "Hierarchical attention based long short-term memory for chinese lyric generation," *Appl. Intell.*, vol. 49, pp. 44-52, 2018, https://doi.org/10.1007/s10489-018-1206-2

[17] C. Park and Y. Cheong, "Automatic rap lyrics generation with rhymes through korean syllables," *SMA*, Jeju, Korea, 2020.

[18] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv2016*, arXiv:1607.04606.

[19] H. Hirjee and D. Brown, "Using automated rhyme detection to characterize rhyming style in rap music," *Empirical Musicology Rev.*, vol. 5, no. 4, pp. 121-145, 2010, https://doi.org/10.18061/1811/48548

[20] P. Potash, A. Romanov, and A. Rumshisky, "Evaluating creative language generation: the case of rap lyric ghostwriting," *arXiv2016*, arXiv:1612.03205v1.

박 찬 솔 (Chansol Park)

2014년 8월 : 단국대학교 멀티미디어공학과 졸업

2018년 3월~현재 : 성균관대학교 전자전기컴퓨터공학과 석사과정

<관심분야> 인공지능, 워드 임베딩, 자연어처리

[ORCID:0000-0002-2198-0445]

정 윤 경 (Yun-Gyung Cheong)

1996년 2월 : 성균관대학교 정보공학과 졸업

1998년 2월 : 성균관대학교 정보공학과 석사

2007년 8월 : 노스캐롤라이나주립대학교 전산학과 박사

2008년~2010년 : 삼성전자 종합기술원 전문연구원

2010년 10월~2014년 8월 : 덴마크 IT University of Copenhagen post-doc

2015년 3월~2017년 3월 : 성균관대학교 소프트웨어대학 조교수

2017년 3월~현재 : 성균관대학교 소프트웨어대학 부교수

<관심분야> 인공지능, 지능적 스토리텔링 및 게임 AI

이 종 현 (Jong-Hyun Lee)

2010년 2월 : 성균관대학교 전자전기컴퓨터공학과 학사

2012년 2월 : 성균관대학교 전자전기컴퓨터공학과 석사

2017년 2월 : 성균관대학교 전자전기컴퓨터공학과 박사

2017년 4월~현재 : 성균관대학교 컨버전스연구소 박사후연구원

<관심분야> 진화 인공지능, 진화 연산, 인공지능 작곡